Robust Mutual Authentication and Key Agreement Protocol for Smart Grid Security

Dissertation submitted to
Visvesvaraya National Institute of Technology, Nagpur
in partial fulfillment of the requirements for the award of
the degree

Master of Technology
in
Computer Science & Engineering

by

CHHAYA HIRWANI (MT23MCS025)

under the guidance of

Dr. Syed Taqi Ali

Department of Computer Science & Engineering

Visvesvaraya National Institute of Technology

Nagpur 440 010 (India)

JUNE 2025

Robust Mutual Authentication and Key Agreement Protocol for Smart Grid Security

Dissertation submitted to
Visvesvaraya National Institute of Technology, Nagpur
in partial fulfillment of the requirements for the award of
the degree

Master of Technology
in
Computer Science & Engineering

by

CHHAYA HIRWANI (MT23MCS025)

under the guidance of

Dr. Syed Taqi Ali

Department of Computer Science & Engineering

Visvesvaraya National Institute of Technology

Nagpur 440 010 (India)

JUNE 2025

©Visvesvaraya National Institute of Technology (VNIT) 2024

Department of Computer Science & Engineering
Visvesvaraya National Institute of Technology, Nagpur

Declaration

I, **Chhaya Hirwani**, hereby declare that this dissertation titled **" Robust Mutual Authentication and Key Agreement Protocol for Smart Grid Security "** is carried out by me in the Department of Computer Science & Engineering of Visvesvaraya National Institute of Technology, Nagpur. The work is original and has not been submitted earlier whole or in part for the award of any degree at this or any other Institution.

Chhaya Hirwani
Name and signature
Date: 06-06-25

Certificate

This to certify that the dissertation titled **" Robust Mutual Authentication and Key Agreement Protocol for Smart Grid Security "** is submitted by **Chhaya Hirwani** in partial fulfillment of the requirements for the award of the degree of **Master of Technology in Computer Science & Engineering**, VNIT Nagpur. The work is comprehensive, complete, and fit for final evaluation.

06-06-25
Dr. Syed Taqi Ali
Department of Computer Science & Engineering, VNIT Nagpur

सहायक प्राध्यापक
Assistant Professor
संगणक विज्ञान एवं अभियांत्रिकी विभाग
Dept. of Computer Science & Engg
वि.रा.प्रौ.सं. नागपुर-10 (भारत)
V.N.I.T., NAGPUR-440010. (India)

Dr. M. P. Kurhekar
Head, Department of Computer Science & Engineering, VNIT Nagpur
Date:

विभाग प्रमुख
Head
संगणक विज्ञान एवं अभियांत्रिकी विभाग
Dept. of Computer Science & Engg.
वि.रा.प्रौ.सं. नागपुर-10 (भारत)
V.N.I.T., Nagpur-440010. (India)

Department of Computer Science & Engineering
Visvesvaraya National Institute of Technology, Nagpur

Non-Plagiarism Certificate

Certified that M. Tech Dissertation titled " **Robust Mutual Authentication and Key Agreement Protocol for Smart Grid Security** " submitted by **Chhaya Hirwani**, Enrollment No. **MT23MCS025**, ID No. **30779** has been checked for plagiarism using TURNITIN software and the overlap is found within prescribed limits. A summary of the report is given below.

ORIGINALITY REPORT

17%	11%	13%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

Chhaya Hirwani
Enrollment No: MT23MCS025

Dr. Syed Taqi Ali 06-06-24
Assistant Professor
Department of Computer Science & Engineering
VNIT, Nagpur

सहायक प्राध्यापक
Assistant Professor
संगणक विज्ञान एवं अभियांत्रिकी विभाग
Dept. of Computer Science & Engg
वि.रा.प्रौ.सं. नागपुर-10 (भारत)
V.N.I.T., NAGPUR-440010. (India)

ACKNOWLEDGEMENT

I would like to express my deepest gratitude and sincere appreciation to all those who have supported and guided me throughout this M.Tech journey.

First and foremost, I am immensely grateful to my supervisor, **Dr. Syed Taqi Ali** Ma'am, for her invaluable guidance, expertise, and encouragement. Her constant support, patience, and insightful feedback have been instrumental in shaping this thesis. I am truly fortunate to have the opportunity to work under her mentorship.

I would like to extend my heartfelt thanks to **Dr. M. P. Kurhekar**, Head of Department, for his invaluable support and guidance throughout this journey. His encouragement has been vital in the successful completion of this project.

I would like to extend my heartfelt thanks to the faculty members of the Department of Computer Science & Engineering at Visvesvaraya National Institute of Technology Nagpur for their knowledge-sharing and for providing a conducive academic environment. I am indebted to them for their teachings and guidance which have broadened my perspective and enhanced my understanding of the subject. My sincere appreciation goes to them for their assistance, collaboration, and providing necessary resources for conducting experiments and carrying out this research work. Their contributions have been significant in the successful completion of this thesis.

I would also like to express my gratitude to my family and friends for their unwavering support, understanding, and encouragement throughout this journey. Their love, motivation, and belief in my abilities have been the driving force behind my accomplishments.

Thank you all.

Chhaya Hirwani

ABSTRACT

Smart meters are a key component of modern smart grids, enabling utility companies to collect and analyze energy consumption data in real time. This capability improves the efficiency of energy distribution and helps consumers better manage their energy usage. However, the real-time nature of this data collection raises significant privacy concerns. If not properly protected, sensitive information about consumers' habits, rou tines, and appliance usage could be exposed. To address these concerns, encrypted aggregation techniques have been widely adopted to ensure the privacy and security of data transmitted by smart meters. In our work, we propose a new solution specifically designed to enhance security and privacy in smart grids. Our approach addresses these challenges by offering a more efficient and feature-rich alternative to existing solutions. By focusing on reducing communication costs, minimizing computational load, and lowering storage requirements, our framework makes smart grid security more practical for real-world applications. We prioritize key security aspects, including mutual authentication, which ensures both the smart meter and the service provider verify each other's identities, protecting against impersonation and other attacks. Our solution also maintains user anonymity, ensuring that power consumption data cannot be linked to specific individuals. Additionally, we protect against common security threats like man-in-the-middle attacks, ensuring that data remains safe during transmission. Our framework provides a robust and secure way to share data within the smart grid while minimizing the resource demands on the system. With its emphasis on both strong security and low overhead, our approach offers a reliable and scalable solution for smart grid communication, contributing to the development of safer, more efficient, and privacy-preserving energy systems. way for safer and smarter energy systems.

TABLE OF CONTENTS

Title **Page No.**

ABSTRACT . . . . . **iv**

TABLE OF CONTENTS . . . . . **v**

LIST OF TABLES . . . . . **viii**

LIST OF FIGURES . . . . . **ix**

1 INTRODUCTION **1**

1.1 Background and Motivation 1

1.2 Problem Statement 3

1.3 Objectives of the Thesis 4

1.4 Key Contributions of the Thesis 4

1.5 Thesis Outline 5

2 LITERATURE SURVEY **7**

2.1 Authentication Techniques 7

2.1.1 Password-based Authentication 7

2.1.2 Biometric-based Authentication 8

2.1.3 Certificate-based Authentication 9

2.1.4 Identity-based Authentication 9

2.1.5 Attribute-based Authentication 11

2.2 Key Agreement Protocols 12

2.2.1 Diffie-Hellman (DH) Key Exchange 12

2.2.2 Elliptic Curve Diffie-Hellman (ECDH) 14
2.2.3 Pre-shared Key (PSK) Methods 15
2.2.4 Ephemeral Key Agreement 16
2.3 Cryptographic Techniques 17
2.3.1 Elliptic Curve Cryptography (ECC) 17
2.3.2 Hash Functions 18
2.3.3 HMAC (Hash-based Message Authentication Code) 18
2.3.4 Lightweight Symmetric Encryption 20
2.3.5 Chaotic Maps and Lightweight Stream Ciphers 20
2.3.6 AES Encryption/Decryption 21
2.3.7 Nonces (Random Numbers Used Once) 21
2.4 Privacy-Preserving Techniques 21
2.4.1 Anonymization 22
2.4.2 Pseudonym-Based Techniques 22
2.4.3 Homomorphic Encryption 23
2.5 Attack Resistance Mechanisms 24
2.5.1 Replay Attack Mitigation 24
2.5.2 Man-in-the-Middle (MitM) Attack Prevention 24
2.5.3 Impersonation Attack: 25
2.5.4 Denial of Service (DoS) Attack Resistance 25
2.5.5 Insider Threat Mitigation 26
2.6 Formal Security Verification Techniques 26
2.6.1 BAN Logic (Burrows-Abadi-Needham Logic): 26
2.6.2 ProVerif and AVISPA Tools: 27
2.6.3 Game-based Security Proofs: 28

3 PROTOCOL DESIGN 30
3.1 Participating Entities 30
3.1.1 Smart Meter (SM) 30
3.1.2 Service Provider (SP) 30
3.1.3 Trusted Authority (TA) or Control Center 31

3.2 Assumptions 32
3.3 Design Goals 32
3.4 Notations 34
3.5 Proposed protocol 34
3.5.1 Registration Phase 35
3.5.2 Authentication and Key Agreement Phase 35
3.6 Protocol Correctness 38
3.6.1 Consistency in Decryption and Computation of r_2 38
3.6.2 Verification of Q_i 39
3.6.3 Consistency in SK Derivation 39
3.6.4 Validation of Authentication Token $Authh$ 39

4 IMPLEMENTATION 40
4.1 Tools and Libraries 40
4.2 Implementation Details 41
4.2.1 Registration Phase 41
4.2.2 Authentication and Key Agreement Phase 41
4.3 Output and Validation 42

5 EXPERIMENTS AND RESULTS 43
5.1 Security Analysis 43
5.1.1 Informal Security Analysis 43
5.1.2 Formal Security Analysis 46
5.2 Performance Analysis 50
5.2.1 Computation Cost 50
5.2.2 Communication Cost 51

6 CONCLUSION AND FUTURE WORK 53
6.1 Conclusion 53
6.2 Future Work 53

List of Tables

3.1 Description of Symbols Used in the Protocol 34
3.2 Registration Phase 35
3.3 Authentication and Key Agreement Phase 36

5.1 Cryptographic Operations 50
5.2 Computation Cost 51
5.3 Communication Cost 52

List of Figures

1.1 Smart Grid scenerio in a Smart City 3

2.1 Authentication phase of the identity-based mutual authentication protocol using PUF [Source: IEEE Xplore, M. S. Badar et al., 2021]. 10
2.2 System diagram for ECDH key exchange between devices with limited resources. Source:[https://doi.org/10.3390/electronics13183631] 14
2.3 Flowchart of the Pre-Shared Key (PSK) Exchange Process 15
2.4 Flowchart of HMAC (Source: https://creately.com/diagram/example/ilkm0z7i1/flow-chart-hmac-classic) 19
2.5 BAN Logic Inference steps (adapted from [Baolong Liu et al.], 2018) . . 27
2.6 AVISPA Architecture (adapted from [Patil Rachna Yogesh et al.], 2020) . 28

3.1 Overview of Registration, Authentication, and Key Establishment Between Smart Meter (SM), Service Provider (SP), and Trusted Authority (TA) . . 31

4.1 Snapshot of protocol execution with computed values and session key . . 42

5.1 Computation Cost 51
5.2 Communication Cost 52

Chapter 1

INTRODUCTION

1.1 Background and Motivation

The traditional power grid system has long served as the backbone of electricity distribution around the world. However, with the increasing demand for energy, the integration of renewable energy sources, and the emergence of advanced digital technologies, a new and more intelligent infrastructure—known as the smart grid—has been developed. Smart grids transform the conventional one-way electricity supply chain into a dynamic, two-way communication network. This modern grid combines power system infrastructure with information and communication technologies (ICT), enabling more efficient energy management, real-time monitoring, automated fault detection, and active participation of consumers in energy usage decisions. Fig. shows Smart Grid scenario in Smart Cities.

A key component of smart grid infrastructure is the smart meter. These devices are installed in homes and businesses to automatically measure electricity usage and communicate that information to utility providers at frequent intervals. This eliminates the need for manual meter readings and provides both the consumer and the provider with detailed consumption data. Such granular data enables better forecasting, peak load management, and the ability to implement flexible pricing models that encourage energy conservation. Furthermore, smart meters contribute significantly to the automation and optimization of grid operations, facilitating quicker outage detection and faster recovery.

Despite these advancements, smart grids and smart meters come with a range of se-

curity and privacy challenges. The continuous exchange of information between the smart meter and the utility company exposes sensitive user data to potential threats. Detailed energy usage patterns can be analyzed to infer when a house is occupied, what appliances are being used, or even the daily routine of the inhabitants. If intercepted or misused, this data could lead to privacy violations, surveillance risks, or targeted criminal activities such as burglary.

Moreover, the open and distributed nature of smart grid communication makes it vulnerable to cyber attacks. Attackers can exploit weaknesses in the communication protocols to perform replay attacks, impersonation, MITM attacks, and data tampering. These attacks can compromise the confidentiality and integrity of the information being exchanged. Even worse, a successful attack could disrupt the entire grid operation, leading to power outages or manipulation of energy bills. As the number of connected devices in the smart grid grows, so does the attack surface, increasing the risk and impact of security breaches.

In addition to these concerns, smart meters are resource limited devices. They have bounded processing power, memory, and energy, making them incapable of handling complex and computationally intensive cryptographic algorithms. Hence, any proposed security mechanism must be lightweight, ensuring strong protection without overwhelming the device's capabilities. Existing solutions either offer strong security at the cost of high resource consumption or are lightweight but vulnerable to attacks. This trade-off presents a significant challenge in the design of authentication and key agreement protocols for smart grid systems.

Motivation : Given the critical role smart meters play in the functioning of smart grids, there is a strong need to develop a secure, lightweight, and privacy-preserving mutual authentication protocol that can work efficiently under constrained resources. The motivation behind this research is to bridge the gap between security and efficiency by designing an authentication mechanism that ensures mutual trust, data confidentiality, and user privacy, while remaining feasible for real-world deployment. A robust and efficient solution not only strengthens smart grid security but also builds consumer confidence, enabling the broader adoption of smart grid technology and contributing to the vision of a secure, reliable, and sustainable energy future.

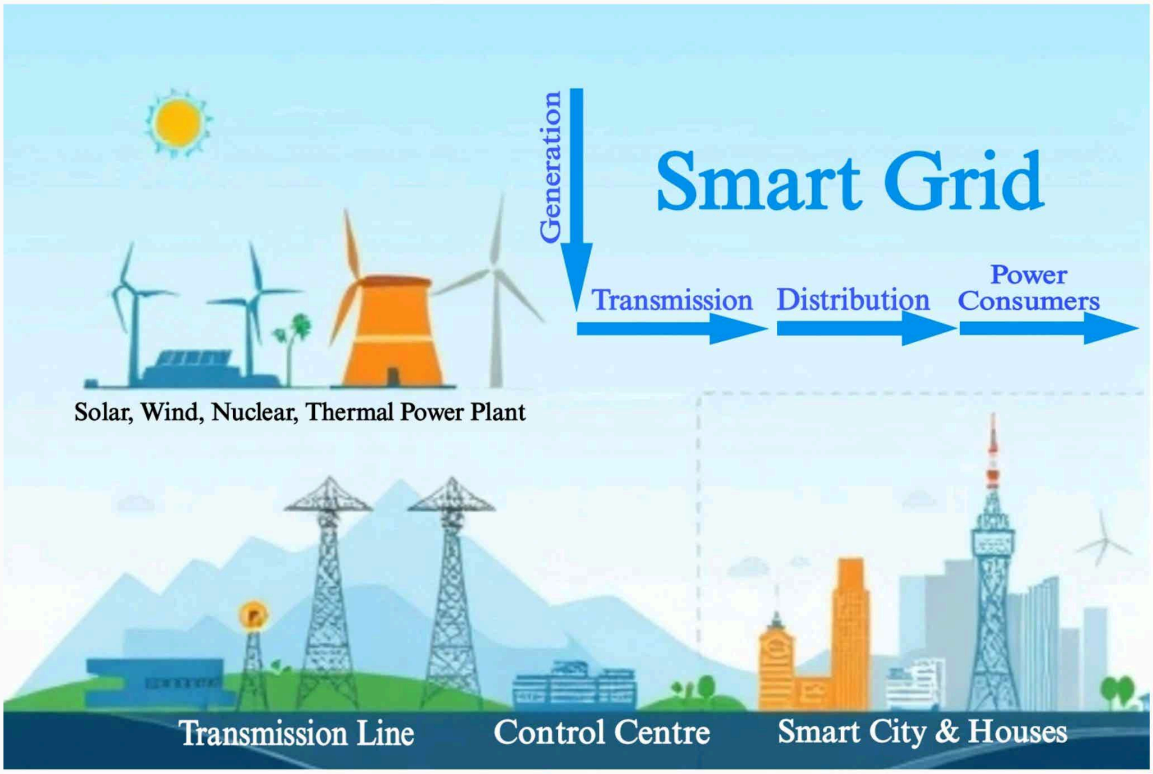

Figure 1.1: Smart Grid scenerio in a Smart City

1.2 Problem Statement

While smart grids and smart meters bring many advantages, they also introduce new challenges related to data security, consumer privacy, and system resilience. Many existing authentication protocols either focus heavily on security but are too complex and resource-intensive for smart meters, or they are lightweight and efficient but lack strong security measures. This leads to several unresolved issues:

1. Lack of mutual authentication, which is needed to verify both the smart meter and the utility provider.
2. Privacy risks from unauthorized access to real-time energy usage data.
3. Weaknesses in defending against cyberattacks, such as replay attacks, man-in-the-middle attacks, and impersonation.
4. Excessive computational and communication overhead that is unsuitable for smart meters with limited processing and memory capabilities.

As a result, there is a clear need for a secure, privacy-preserving, and efficient authentication mechanism that can work effectively in smart grid systems without overburdening the devices involved.

1.3 Objectives of the Thesis

The major objective of this thesis is to propose a new mutual authentication and key agreement protocol that guarantees secure and reliable communication between smart meters and service providers. The specific objectives are:

1. To analyze the security and efficiency gaps in existing smart grid authentication schemes.
2. To design a mutual authentication protocol that ensures both parties (smart meter and service provider) are verified before any data exchange
3. To protect user privacy by anonymizing data and preventing unauthorized access to consumption patterns.
4. To ensure data integrity and confidentiality by using secure cryptographic techniques.
5. To make the protocol resistant to known attacks, such as replay attacks, man-in-the-middle attacks, and insider threats.
6. To optimize the protocol to run efficiently on low-resource devices like smart meters by reducing computation, communication, and storage costs.
7. To evaluate the protocol's performance through theoretical analysis and compare it with existing schemes.

1.4 Key Contributions of the Thesis

This thesis contributes to the field of smart grid security by proposing a novel protocol that is secure, lightweight, and practical. The key contributions are:

- A new authentication and key agreement protocol specifically designed for smart grids, which uses lightweight encryption, hash functions, HMACs, and nonce-based mechanisms to ensure security with minimal overhead.
- The proposed protocol supports mutual authentication, ensuring that both the smart meter and the service provider verify each other's identity before exchanging data.
- The scheme ensures privacy protection by anonymizing the identity of the consumer during data transmission, thereby preventing data linkability and unauthorized tracking.
- It offers resistance to differnt attacks, including replay attacks, man-in-the-middle attacks, impersonation, and insider threats.
- The protocol is optimized for performance, requiring low computational power and communication bandwidth, which makes it suitable for resource limited devices such as smart meters.
- A comprehensive comparison with existing schemes is conducted, showing that the proposed method achieves enhanced trade-offs between security and efficiency.
- The design considers real-world applicability, meaning it can be implemented in current and future smart grid systems with minimal modifications.

By addressing key privacy and security challenges, this work supports the safe and efficient operation of smart grids, building trust among consumers and supporting a sustainable energy future.

1.5 Thesis Outline

There are six chapters in this thesis. The research background, problem statement, goals, and major contributions pertaining to secure communication in smart grid systems are presented in Chapter 1. A literature review of current authentication and key agreement protocols is given in Chapter 2. The suggested protocol design is described in detail in Chapter 3, along with the system model, underlying presumptions, design objectives,

and the protocol's sequential operation. The implementation is covered in detail in Chapter 4, which also includes descriptions of the development environment, tools, and sample outputs. Chapter 5 assesses the protocol's performance in terms of computation and communication overhead and presents the security analysis using formal techniques such as BAN logic. Chapter 6 wraps up the thesis by providing a summary of the findings and outlining potential avenues for further investigation and improvement.

Chapter 2

LITERATURE SURVEY

In this chapter, we present a detailed review of the existing techniques and research related to the security of smart grid communications. We focus on various methods used to protect consumer data, establish secure communication, and prevent unauthorized access. These include authentication mechanisms, key agreement protocols, cryptographic techniques, privacy-preserving strategies, attack-resistant models, and formal security verification approaches. By studying these works, we seek to analyze the strengths and weaknesses of each and identify the gaps that our proposed protocol will address.

2.1 Authentication Techniques

Authentication ensures that only legitimate users or devices can access the smart grid system. Many researchers have proposed different ways to verify the identity of smart meters and service providers. Some used passwords and IDs, while others used smart cards, biometrics, or digital signatures.

2.1.1 Password-based Authentication

The simplest and most popular approach is password-based authentication. This method involves a user or device providing a secret key or predefined password, which the system then checks to confirm the identity. It needs few resources and is easy to implement. In their protocol, [1], [2] employed this authentication method.

Password-based authentication is a handy option for many systems since it is easily implemented and generally known. It keeps expenses low and setup basic by not requiring sophisticated hardware or specialized infrastructure. This makes it especially fit for small-scale projects or settings where top-level security is not a main issue.

Password-based systems have major security restrictions, though. Many times, users choose easily guessed-upon or cracked weak or commonly used passwords. Dictionary attacks using common or predictable passwords and brute-force attacks—where attackers methodically try all possible combinations—both of which highly expose these systems. Password-based authentication is not best for systems needing high security guarantees because of these flaws.

2.1.2 Biometric-based Authentication

This method makes use of individual physical or behavioral traits of a person, such fingerprints, facial traits, iris patterns, or voice. Since these features are difficult to replicate or create, biometric authentication guarantees that only the real user may access the system. [2], [8] decided to apply this method in their procedure.

The level of security offered by biometric-based authentication is substantially higher than that of conventional password-based techniques. It lowers the possibility of weak or reused credentials by doing away with the need for users to remember or manage passwords. This approach provides robust protection against impersonation and unauthorized access because each person's biometric characteristics—such as fingerprints, facial features, or iris patterns—are distinct and challenging to duplicate.

Biometric authentication has certain drawbacks despite its advantages. It calls for specific hardware, such as cameras or fingerprint scanners, which might not be feasible for gadgets with limited resources, like smart meters. Furthermore, because biometric data is permanent and sensitive, using it presents significant privacy issues. Unlike passwords, biometric information cannot be reset or altered if it is ever compromised. Furthermore,

it is still very difficult to transmit and store biometric data securely, so strong security measures are crucial.

2.1.3 Certificate-based Authentication

Digital certificates and a Public Key Infrastructure (PKI) underlie certificate-based authentication. Every system entity has a digital certificate produced by a reputable certificate authority (CA). These certifications support in verifying the identity and in creating safe communication like used in [3].

By utilizing cryptographic techniques, certificate-based authentication provides strong security. It creates secure communication channels and authenticates users using digital certificates, which are hard to fake. This technique, which offers a reliable and standardized approach to authentication and encryption, is extensively used in many secure systems, including HTTPS for websites and online banking.

However, maintaining the Public Key Infrastructure (PKI) and certificates can be difficult and resource-intensive. Lightweight devices like smart meters might not be able to handle tasks like certificate generation, verification, and revocation because they demand a lot of memory and processing power. The system's total overhead and maintenance complexity are further increased by the need for constant monitoring and updates to manage certificate lifecycles, including renewal and revocation.

2.1.4 Identity-based Authentication

Under identity-based authentication (IBA), a user's public key is derived straight from their identity, say an email address or a distinct user ID. This streamlines key management and replaces the necessity for digital certificates. used in [9].

For ex. Three main entities comprise the authentication phase of the identity-based mutual authentication protocol as illustrated in [4] and Fig. 2.1 the Smart Vehicle (SV), the Gateway Node (GWN), and the Control Center (CC). This phase guarantees between communicating parties safe session key creation and mutual authentication. The SV first

computes authentication data using its identity and a one-time token derived from a Physical Unclonable Function (PUF), then creates random nonces. It delivers this information to the GWN, which checks the message and forward it together with its own authentication data to the CC. The CC then checks the validity of the SV and GWN by means of several verifications conducted using the kept identities and tokens. After effective confirmation, the CC responds with a safe message enabling all sides to derive a shared session key. Freshness, identity privacy, and resistance against repeat and man-in—-the-middle attacks are guaranteed by this step-wise message exchange system. PUFs offer unclonable device-specific identities, so improving security even more.

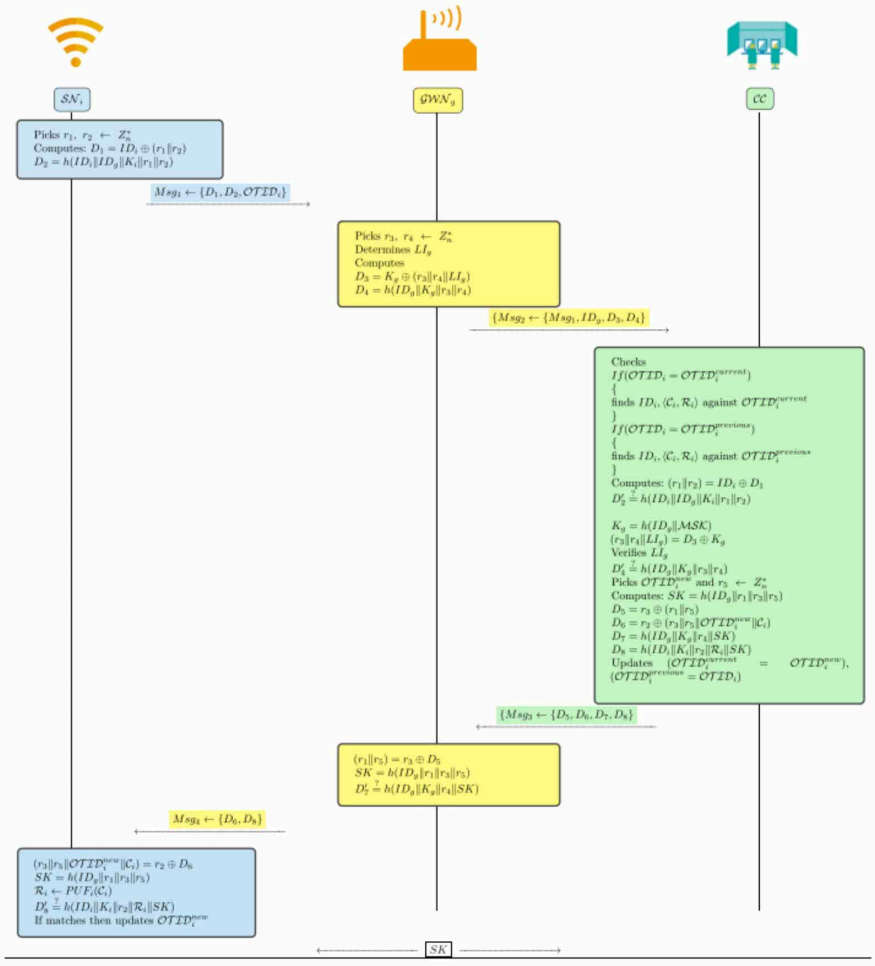
SN_i
GWN_g
CC
Picks $r_1,\ r_2 \leftarrow Z_n^*$
Computes: $D_1 = ID_i \oplus (r_1 \| r_2)$
$D_2 = h(ID_i \| ID_g \| K_i \| r_1 \| r_2)$
$Msg_1 \leftarrow \{D_1, D_2, \mathcal{OTID}_i\}$
Picks $r_3,\ r_4 \leftarrow Z_n^*$
Determines LI_g
Computes
$D_3 = K_g \oplus (r_3 \| r_4 \| LI_g)$
$D_4 = h(ID_g \| K_g \| r_3 \| r_4)$
$\{Msg_2 \leftarrow \{Msg_1, ID_g, D_3, D_4\}$
Checks
$If (\mathcal{OTID}_i = \mathcal{OTID}_i^{current})$
{
finds $ID_i, \langle \mathcal{C}_i, \mathcal{R}_i \rangle$ against $\mathcal{OTID}_i^{current}$
}
$If (\mathcal{OTID}_i = \mathcal{OTID}_i^{previous})$
{
finds $ID_i, \langle \mathcal{C}_i, \mathcal{R}_i \rangle$ against $\mathcal{OTID}_i^{previous}$
}
Computes: $(r_1 \| r_2) = ID_i \oplus D_1$
$D_2' \stackrel{?}{=} h(ID_i \| ID_g \| K_i \| r_1 \| r_2)$
$K_g = h(ID_g \| MSK)$
$(r_3 \| r_4 \| LI_g) = D_3 \oplus K_g$
Verifies LI_g
$D_4' \stackrel{?}{=} h(ID_g \| K_g \| r_3 \| r_4)$
Picks $\mathcal{OTID}_i^{new}$ and $r_5 \leftarrow Z_n^*$
Computes: $SK = h(ID_g \| r_1 \| r_3 \| r_5)$
$D_5 = r_5 \oplus (r_1 \| r_3)$
$D_6 = r_2 \oplus (r_3 \| r_5 \| \mathcal{OTID}_i^{new} \| \mathcal{C}_i)$
$D_7 = h(ID_g \| K_g \| r_4 \| SK)$
$D_8 = h(ID_i \| K_i \| r_2 \| \mathcal{R}_i \| SK)$
Updates $(\mathcal{OTID}_i^{current} = \mathcal{OTID}_i^{new})$,
$(\mathcal{OTID}_i^{previous} = \mathcal{OTID}_i)$
$\{Msg_3 \leftarrow \{D_5, D_6, D_7, D_8\}$
$(r_1 \| r_3) = r_3 \oplus D_5$
$SK = h(ID_g \| r_1 \| r_3 \| r_5)$
$D_7' \stackrel{?}{=} h(ID_g \| K_g \| r_4 \| SK)$
$Msg_4 \leftarrow \{D_6, D_8\}$
$(r_3 \| r_5 \| \mathcal{OTID}_i^{new} \| \mathcal{C}_i) = r_2 \oplus D_6$
$SK = h(ID_g \| r_1 \| r_3 \| r_5)$
$\mathcal{R}_i \leftarrow PUF_i(\mathcal{C}_i)$
$D_8' \stackrel{?}{=} h(ID_i \| K_i \| r_2 \| \mathcal{R}_i \| SK)$
If matches then updates $\mathcal{OTID}_i^{new}$
SK

Figure 2.1: Authentication phase of the identity-based mutual authentication protocol using PUF [Source: IEEE Xplore, M. S. Badar et al., 2021].

By doing away with the requirement for certificates, identity-based authentication greatly lowers overhead and streamlines the authentication procedure. This makes it par-

ticularly appropriate for settings like smart grids that have a lot of users or devices. The system is made lighter and more effective by using a user's identity (such as an email address or device ID) as the public key. This allows for quicker authentication without the hassle of managing certificates.

Nevertheless, this method mainly relies on a reliable entity that creates private keys using user identities. The honesty and reliability of this central authority are critical to the system's overall security. All private keys might be exposed if the authority is compromised, endangering the system as a whole. Consequently, identity-based authentication presents a crucial point of vulnerability even though it is straightforward and effective.

2.1.5 Attribute-based Authentication

Under attribute-based authentication (ABA), users are authenticated depending on a set of criteria including their role (e.g., operator, technician), department, geographic location, or time of access. Systems needing fine-grained access control make extensive use of it.

Designed for mobile environments, [5] put forth an attribute-based authentication (ABA) system. This protocol uses user attributes—such as role, location, or clearance level—instead of depending just on identity-based credentials to enforce fine-grained access control and authentication. The system guarantees that users may be authenticated depending on particular attribute sets without disclosing their actual identity, so improving privacy and flexibility. While remaining efficient for limited devices, security analysis and performance results show that this approach resists repeat, impersonation, and collusion attacks.

By analyzing particular attributes like user roles, locations, or access times, attribute-based authentication provides comprehensive and adaptable control over user access. This enables systems to implement intricate access control rules, such as limiting access to maintenance data to working hours or limiting device interaction to technicians from a

particular area. Because of its adaptability, it is especially useful in dynamic settings like smart grids, where access needs may fluctuate regularly.

This strategy does have some drawbacks, though. With the increase in the number of users and rules, it can become more difficult to manage user attributes and enforce access policies. To avoid manipulation or abuse, attribute data must also be transmitted and stored securely. Delays may also be introduced by the requirement to confirm attributes and policies at each access request, which could be problematic for systems that need quick or real-time response times.

2.2 Key Agreement Protocols

Particularly in systems like smart grids, key agreement protocols are absolutely essential in safe communication. These systems let two or more people agree on a shared secret key, which is subsequently used for communication's encryption and decryption. Strong key agreement systems guarantee that the key stays safe even in cases of exposure of some communication data.

Key agreement protocols in smart grid systems—where devices like smart meters and home gateways sometimes have limited computational power and memory—must be both lightweight and secure. There are many ways to create secret keys. The most often used key agreement strategies are those below:

2.2.1 Diffie-Hellman (DH) Key Exchange

Diffie-Hellman is one of the oldest and most well-known methods for key exchange. It permits two parties to jointly create a confidential key through a public channel. without sharing the key directly. Protocols like [6] have used this method. Below Shows how it works :

Step 1: Public Parameters (agreed upon by both parties)

A large prime number:p

A primitive root modulo p: g (also called the base).

These values p and g are public and can be known to everyone.

Step 2: Private Key Selection

Alice selects a private key

$$a, \quad \text{where } 1 < a < p$$

Bob selects a private key

$$b, \quad \text{where } 1 < b < p$$

Step 3: Public Key Computation

Alice computes her public key:

$$A = g^a \bmod p$$

Bob computes his public key:

$$B = g^b \bmod p$$

These public keys A and B are exchanged over the public channel.

Step 4: Shared Secret Key Computation

Alice computes the shared key:

$$K = B^a \bmod p$$

Bob computes the shared key:

$$K = A^b \bmod p$$

Since:

$$K = g^{ab} \bmod p$$

Both parties generate an identical secret key K, without ever transmitting it.

By letting two people create a shared secret key without really distributing the key over the network, Diffie-Hellman key exchange offers great security. Through preventing

the theft of stationary keys, this dynamic key generating mechanism improves security. Considered a reliable and established method in cryptographic applications, it has been extensively embraced in safe communications.

Still, the approach has certain drawbacks. It depends on intricate mathematical computations including modular exponentiation, which can be computationally demanding and inappropriate for devices with limited resources like smart meters. Furthermore vulnerable to man-in—the-middle attacks whereby an attacker may intercept and control the key exchange process is Diffie-Hellman if it is not coupled with appropriate authentication methods.

2.2.2 Elliptic Curve Diffie-Hellman (ECDH)

ECDH is a modern and further efficient version of the traditional Diffie-Hellman protocol. It is based on Elliptic Curve Cryptography (ECC), which achieves the same level of security with much smaller key sizes. [7] and below Fig. 2.2 illustrates how key negotiation occurs among devices with limited resources.

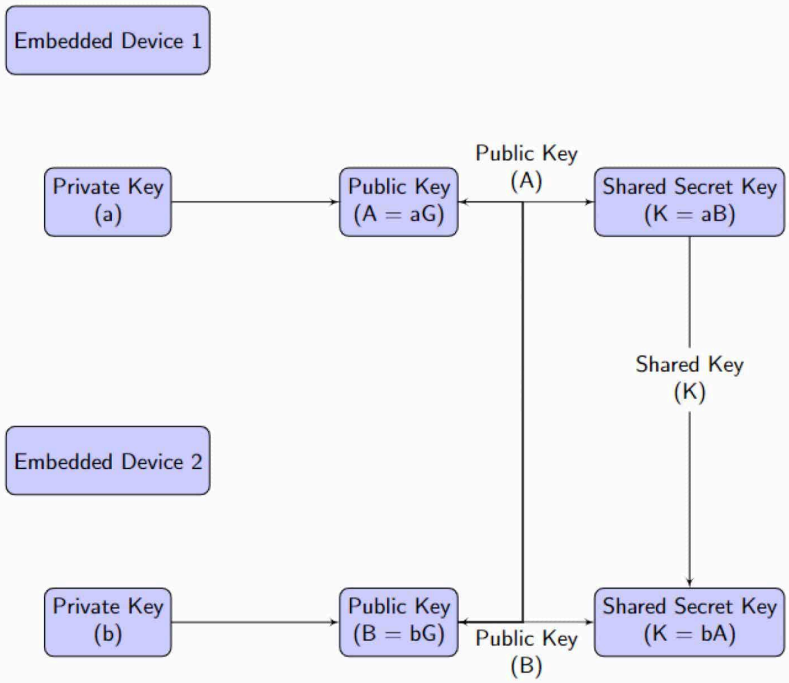

Figure 2.2: System diagram for ECDH key exchange between devices with limited resources. Source:[https://doi.org/10.3390/electronics13183631]

Because Elliptic Curve Diffie-Hellman (ECDH) is more lightweight and efficient than traditional Diffie-Hellman, it presents a number of benefits. For limited environments like smart meters, IoT devices, and embedded systems, it is perfect since it calls much less computational power and memory. Strong security guaranteed by ECDH also helps to lower energy consumption and bandwidth usage—two crucial considerations in low-power and resource- constrained systems even with shorter key lengths.

ECDH has certain restrictions notwithstanding its benefits. Maintaining security still depends on a safe and deliberate application of elliptic curve operations. The quality and dependability of the ECC (Elliptic Curve Cryptography) library applied determines mostly the security and performance. Furthermore, both the communicating parties have to agree on the elliptic curve parameters ahead, which could call for system compatibility and coordination.

2.2.3 Pre-shared Key (PSK) Methods

In PSK-based methods, both communicating parties already share a secret key before the communication begins. This key is used directly for authentication and encryption. Below flowchart in shows the procedure of PSK.

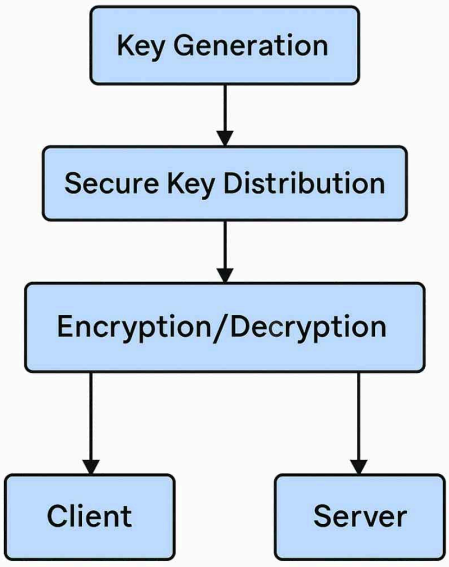

Figure 2.3: Flowchart of the Pre-Shared Key (PSK) Exchange Process

Pre-shared key (PSK) authentication has the benefits of being rather quick and easy to apply. It is lightweight and appropriate for closed or small-scale systems whereby devices are manually pre-configured with the key since it does not depend on sophisticated mathematical operations or public-key cryptography. PSK is therefore perfect in settings when simplicity and speed take front stage.

PSK does, nevertheless, also have certain clear restrictions. Both managing and updating keys in big or dynamic networks can be challenging and unworkable. Should a pre-shared key be compromised, all communications under protection become exposed, so causing a total collapse of security. Furthermore, if appropriate authentication mechanisms are not stacked on top of this approach, it can be vulnerable to impersonation attacks and does not scale well with growing numbers of consumers or devices.

2.2.4 Ephemeral Key Agreement

Ephemeral key agreement techniques use temporary (short-lived) keys for each session, ensuring that even if one session key is compromised, other sessions remain secure. This property is called forward secrecy.

Because each session generates a new key, ephemeral key agreement strengthens communication security. This method guarantees forward secrecy, thus previous communications remain safe and cannot be deciphered even in case a current session key is hacked. Since every key is only temporary and used just once, it also reduces the long-term effects of a major leak.

Ephemeral key agreement brings some difficulties notwithstanding these benefits. Frequent key generation required in the process might somewhat raise computational overhead, especially on devices with limited resources. The temporary keys also have to be swapped and controlled safely. The system stays susceptible to man-in—-middle attacks during the key sharing process without appropriate authentication systems.

2.3 Cryptographic Techniques

Cryptography is the backbone of secure communication. It helps in keeping information confidential, ensuring it has not been changed, and verifying the identity of the sender. In smart grid environments, which include devices like smart meters and home gateways with limited memory and processing power, cryptographic methods must be lightweight, efficient, and secure. Below are some important cryptographic techniques commonly used in smart grid communication:

2.3.1 Elliptic Curve Cryptography (ECC)

ECC is a form of asymmetric cryptography, which means it uses a public and private key pair. It is widely used in modern systems such as in papers like [2], [8] because it provides strong security with smaller keys compared to traditional methods like RSA. Elliptic Curve Cryptography (ECC) is based on the elliptic curve equation of the form:

$$y^2 = x^3 + ax + b \mod p$$

where:

- a, b are constants,
- p is a prime number defining the finite field $\mathbb{F}_p$,

The curve must satisfy the condition:

$$4a^3 + 27b^2 \neq 0$$

to ensure that the curve has no singularities (i.e., it is non-singular).

Smart meters and IoT devices are among the environments with limited resources where Elliptic Curve Cryptography (ECC) is quite helpful. Comparatively to conventional public-key cryptographic techniques like RSA, it calls much less processing power, memory, and bandwidth. ECC offers strong security elements including encryption, digital

signatures, and key exchange even if it is light weight. This makes it a sensible and effective way to guard communication in limited systems without sacrificing cryptographic strength.

Example use:

- In the context of mutual authentication involving the smart meter and the service provider, ECC can be used to securely agree on a secret key.

2.3.2 Hash Functions

Hash functions take input data and produce a fixed-length output, known as a digest. They are used to ensure data integrity and support fast, lightweight authentication.

Popular hash functions:

1. SHA-256 – produces a 256-bit hash and is widely used for security.
2. HMAC – a type of keyed hash function used for message authentication.

Fast, simple to use, and vital for maintaining data integrity, hash functions are therefore rather handy. By creating a fixed-size output (hash) from variable-length input data, they assist to confirm that data has not been changed during transmission. Even a minor modification in the input produces a totally different hash, which makes manipulation readily visible. Essential for many systems' safe communication and verification systems, hash functions are also extensively applied in device authentication and in creating digital signatures.

2.3.3 HMAC (Hash-based Message Authentication Code)

A secret key and a cryptographic hash function are combined in the HMAC technique to guarantee the authenticity and integrity of a message. Secure communication protocols like TLS, IPsec, and HTTPS make extensive use of it.

HMAC Formula

The HMAC for a message M using a key K and a hash function H is computed as:

$$\text{HMAC}_K(M) = H\left((K' \oplus \text{opad}) \parallel H((K' \oplus \text{ipad}) \parallel M)\right)$$

Where:

- K' is the key K padded to the block size of the hash function,
- $\oplus$ denotes bitwise XOR,
- $\parallel$ denotes concatenation,
- `ipad` is the inner padding (0x36 repeated to block size),
- `opad` is the outer padding (0x5c repeated to block size).

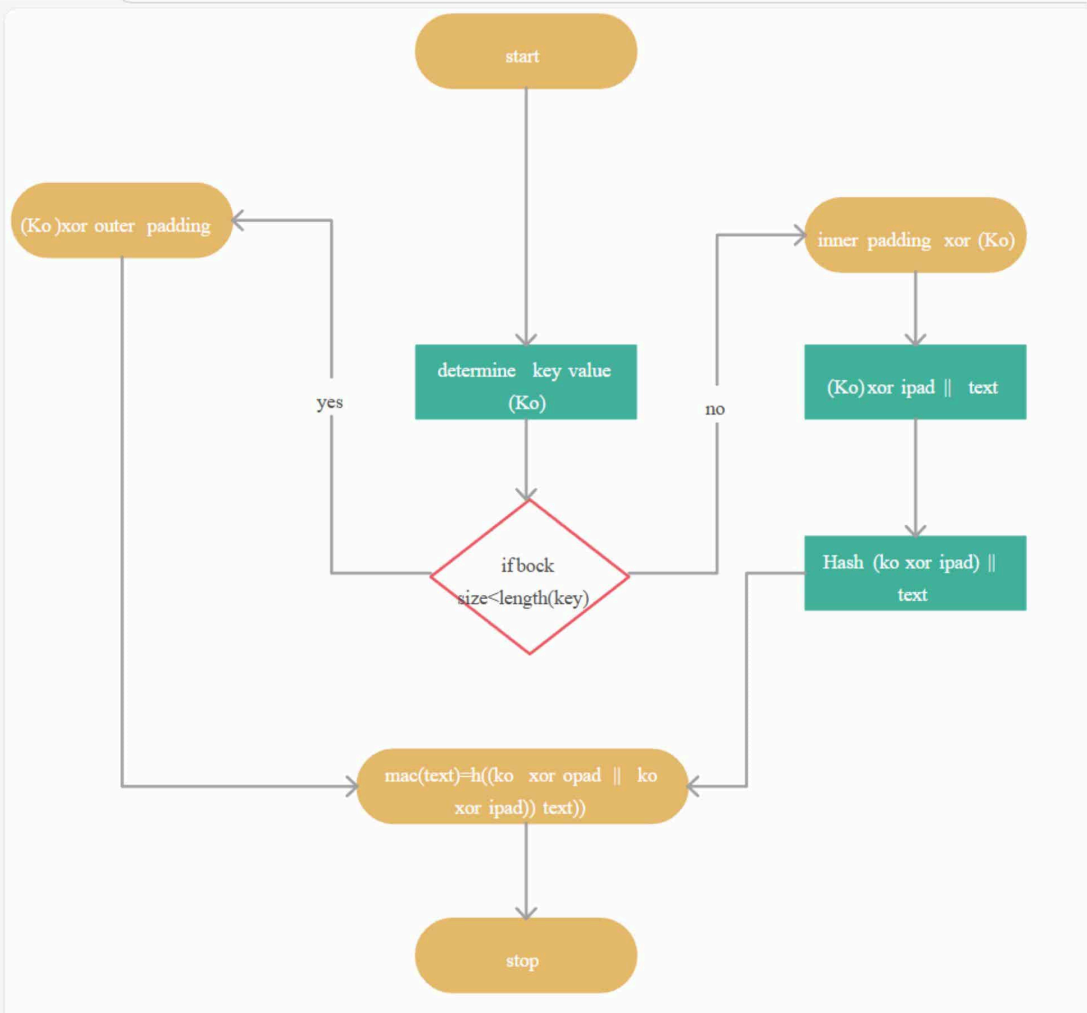

Figure 2.4: Flowchart of HMAC (Source: https://creately.com/diagram/example/ilkm0z7i1/flow-chart-hmac-classic)

2.3.4 Lightweight Symmetric Encryption

Symmetric encryption uses the same key for both encryption and decryption. It is very fast and well-suited for real-time communication between smart grid devices.

Common lightweight symmetric ciphers:

- DES (Data Encryption Standard) – older but still used in simple systems.
- AES (Advanced Encryption Standard) – especially AES-128, which is secure and efficient.

Lightweight symmetric encryption is incredibly useful because it offers a much faster way to secure data compared to public key encryption. Since it requires less memory and has lower CPU demands, it's perfect for devices with limited resources. This makes it an ideal choice for encrypting messages between smart meters and servers, where efficiency and speed are crucial. By using lightweight symmetric encryption, communication remains secure without overburdening the devices, ensuring smooth and reliable data exchange in smart grid systems.

2.3.5 Chaotic Maps and Lightweight Stream Ciphers

These are newer cryptographic techniques being explored for smart grids and IoT applications.

Chaotic maps:

Chaotic maps create intricate and unpredictable sequences that are perfect for encryption by utilizing ideas from mathematical chaos theory. Because of the high degree of randomness introduced by these sequences, attackers find it challenging to anticipate or replicate the encryption pattern. Chaotic maps also have a lightweight design, which allows them to offer robust security without consuming a lot of processing power. Chaotic maps are a useful tool in contemporary encryption techniques because of their efficiency and complexity balance.

Lightweight stream ciphers:

One bit or byte at a time, lightweight stream ciphers encrypt data allowing quick and effective processing. They are ideal for devices with limited resources since they use little keys and demand low computational capability. Particularly crucial in applications like smart meters and other IoT devices, this simplicity and efficiency allow safe communication without taxing the system.

Because they offer an alternative to traditional cryptographic techniques in cases where they might be too slow or resource-intensive, lightweight stream ciphers are crucial. They are particularly useful for devices with little memory or processing power because of their effectiveness and low hardware requirements. Furthermore, researchers are still working on lightweight stream ciphers as they create ever-lighter cryptographic methods to satisfy the expanding demands of new applications and technologies.

2.3.6 AES Encryption/Decryption

AES works in fixed-size blocks (usually 128 bits) and uses keys of 128, 192, or 256 bits. In smart grid systems, AES-128 is commonly used because it balances strong security with efficient processing. It's great for protecting stored data and for secure communication between devices.

2.3.7 Nonces (Random Numbers Used Once)

A nonce is a unique random number used in protocols to make sure that a message or request is not reused. Including a nonce in a message helps prevent replay attacks and ensures that every session is fresh and different. Nonces are simple, but very effective for security.

2.4 Privacy-Preserving Techniques

Privacy is a major concern in smart grid communication. Devices like smart meters continuously collect and send data about electricity usage. If not properly protected, this

information could reveal personal habits, routines, or even when someone is at home. Therefore, smart grid systems need techniques that protect user identity and data privacy while still allowing necessary communication.

Below are some commonly used privacy-preserving methods in authentication and key agreement protocols:

2.4.1 Anonymization

Anonymization is the process of eliminating or hiding personal identifiers (such as user ID, name, or address) from the data being sent.

Because anonymization keeps hackers from connecting usage data to a particular individual, it helps protect people. Anonymization guarantees that sensitive information stays confidential and cannot be linked to a specific person, even in the event that the communication is intercepted. Maintaining security and trust requires this extra layer of privacy, particularly in settings where personal information is regularly transferred.

Common methods:

- Masking or removing identity fields.
- Sending only the necessary information without user-specific details.

One of the main drawbacks of anonymizing is that it has to be used carefully to prevent inadvertently compromising system integrity or upsetting system functionality. Anonymizing poorly can leave holes that attackers could use or cause the system to lose critical functions, so reducing the efficacy or dependability of the technique. Thus, when using anonymizing methods, it is imperative to find the proper mix between privacy and performance.

2.4.2 Pseudonym-Based Techniques

Instead of using real user identities, these methods assign fake or temporary identifiers (called pseudonyms) to users or devices. These pseudonyms are changed regularly.

Pseudonym-based methods are crucial since they enhance privacy by helping external observers not follow users over the long run. They simultaneously enable the system to link pseudoms back to actual users when needed, so balancing anonymity with responsibility. This guarantees that there is still a means to uphold security and responsibility inside the system even while users value continuous surveillance protection.

Finding a safe way to manage and update pseudoms without sacrificing the general integrity of the system presents a significant obstacle with pseudonym-based approaches. Improper handling of pseudonyms can result in weaknesses compromising user privacy or system functionality disturbance. Maintaining security and confidence depends on safe and seamless pseudonym updates.

Example:

- A smart meter might send data using a different pseudonym each time, making it hard for attackers to track its activities over time.

2.4.3 Homomorphic Encryption

It is a powerful technique that enables computations on encrypted data without the need for decryption. For example, a server can add two encrypted values and get the correct encrypted result, which can later be decrypted by the user.

Homomorphic encryption is important since it lets data remain private and confidential while yet enabling helpful processing chores like billing or data analysis. Sensitive material can thus be securely handled all through its lifetime without ever revealing the raw data.

Its primary drawback, though, is that it consumes a lot of computational capability, which makes it overly weighty for devices with limited resources like smart meters. Homomorphic encryption is thus mostly used in cloud systems or environments with more computational capacity where its advantages can be completely utilized.

Relevance:

- Although not yet widely used in lightweight systems, it is a promising direction for future smart grid privacy solutions.

2.5 Attack Resistance Mechanisms

Smart grid systems are constantly at risk of cyber-attacks due to their connected nature and the sensitive data they handle. It is essential for mutual authentication and key agreement protocols to include features that can resist common attacks and maintain system reliability and user trust.Below are the main types of attacks and the typical solutions used to resist them:

2.5.1 Replay Attack Mitigation

A replay attack occurs when an attacker intercepts and captures a message, then resends it to the system to trick it into treating the message as a legitimate user request or as a new valid request. To prevent this, the system can use nonces (random numbers used only once) or timestamps in each communication session. These values help the system verify that each message is fresh and not a reused or duplicated one, effectively blocking replay attempts.

Example:

Smart meter includes a fresh timestamp or random number in its authentication message. If the same message is sent again, it will be rejected.

2.5.2 Man-in-the-Middle (MitM) Attack Prevention

A man-in-the-middle attack happens when someone secretly listens in on a conversation between two parties—and might even change the messages without either side knowing. This can lead to data theft or unauthorized access. To defend against this, systems should use mutual authentication—where both parties verify each other's identity—and

encrypted key exchange protocols, so that even if messages are intercepted, the attacker cannot understand or alter the data.

Example:

The smart meter and service provider both authenticate using public-private key techniques, and the communication is encrypted end-to-end.

2.5.3 Impersonation Attack:

In an impersonation attack, the attacker pretends to be a legitimate user by stealing or faking credentials. The system can prevent this by using strong identity verification techniques, such as biometric authentication or secure cryptographic credentials. This guarantees that only users with proper authorization and valid credentials can access the system.

Example:

A smart meter may use biometric keys or encrypted identity values that are difficult for attackers to forge.

2.5.4 Denial of Service (DoS) Attack Resistance

A DoS attack tries to overwhelm the system with too many fake requests, making it unavailable for legitimate users. To avoid this, systems can implement lightweight challenge-response mechanisms or pre-authentication filters that quickly detect and reject suspicious or unnecessary traffic before it consumes too many resources.

Example:

The server checks for a valid nonce or quick response before processing full authentication, saving time and resources.

2.5.5 Insider Threat Mitigation

Insider threats come from users who already have access to the system but misuse it intentionally. These are harder to detect. A good defense is to use multi-factor authentication (like combining passwords with biometrics or smart cards) and data auditing, which keeps track of user actions so any misuse can be identified and stopped quickly.

Example:

Even if a service provider employee has login credentials, the system may require a second factor like a fingerprint scan and keep logs of all actions taken.

2.6 Formal Security Verification Techniques

To ensure that authentication and key agreement protocols are truly secure, researchers use formal methods to analyze and verify them. These techniques help in proving that the protocols are resistant to different types of attacks and behave as expected under all conditions.

2.6.1 BAN Logic (Burrows-Abadi-Needham Logic):

This is a reasoning method used to analyze the security of authentication protocols. It helps determine whether the participants (like the smart meter and service provider) can truly believe they are communicating securely with the intended party. BAN logic checks whether messages can be trusted and whether beliefs about key ownership and freshness are valid.

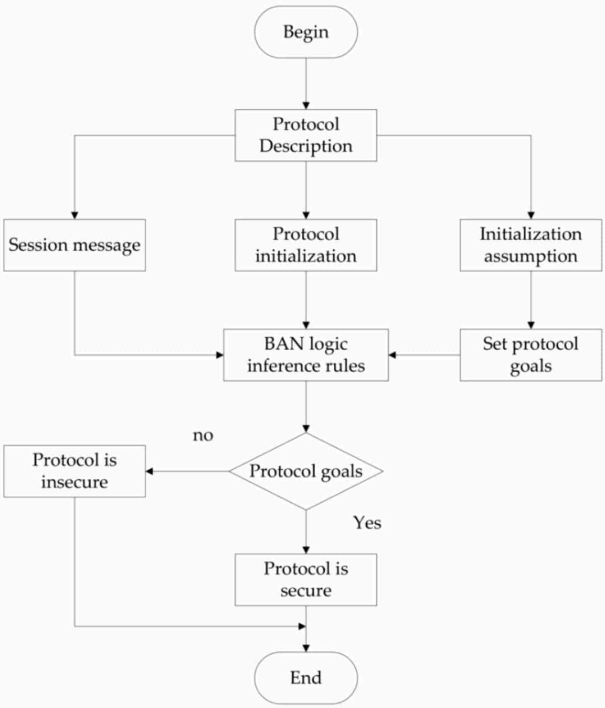

Figure 2.5: BAN Logic Inference steps (adapted from [Baolong Liu et al.], 2018)

2.6.2 ProVerif and AVISPA Tools:

These are automated tools that perform formal analysis of security protocols. ProVerif can model cryptographic operations and verify properties like secrecy and authentication. AVISPA (Automated Validation of Internet Security Protocols and Applications) is another tool that checks whether the protocol is vulnerable to known attacks. These tools are widely used in research for verifying smart grid protocols.

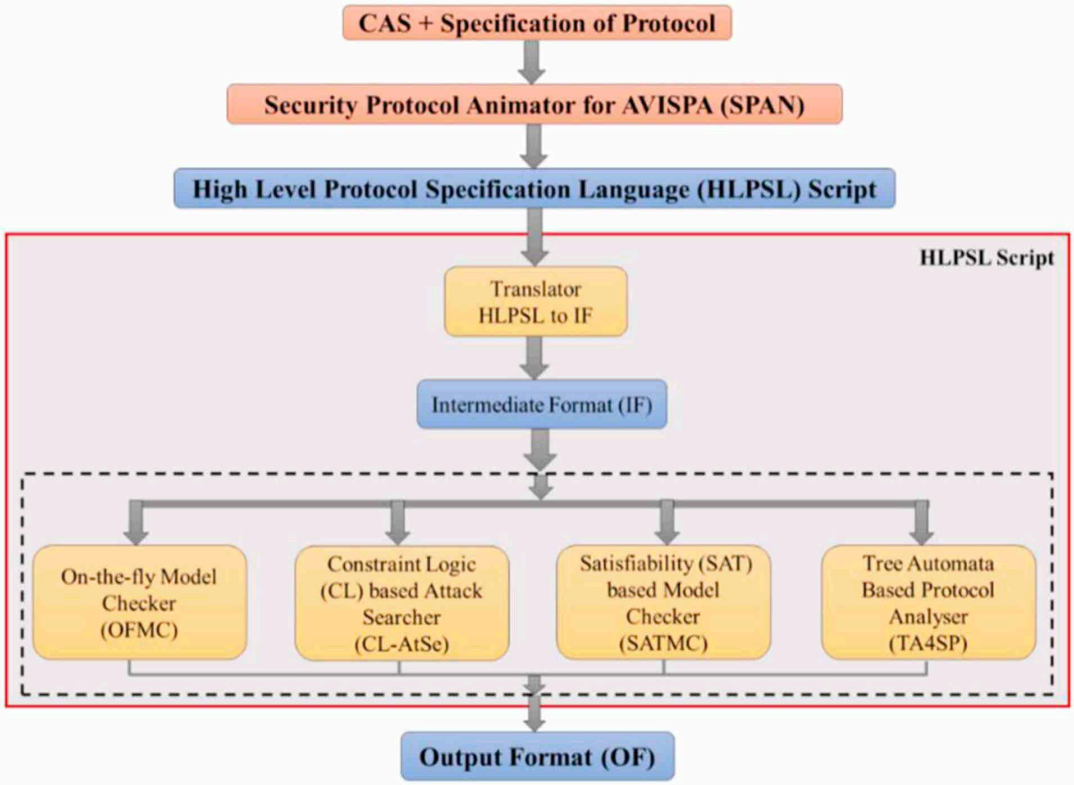
CAS + Specification of Protocol
Security Protocol Animator for AVISPA (SPAN)
High Level Protocol Specification Language (HLPSL) Script
HLPSL Script
Translator
HLPSL to IF
Intermediate Format (IF)
On-the-fly Model Checker (OFMC)
Constraint Logic (CL) based Attack Searcher (CL-AtSe)
Satisfiability (SAT) based Model Checker (SATMC)
Tree Automata Based Protocol Analyser (TA4SP)
Output Format (OF)

Figure 2.6: AVISPA Architecture (adapted from [Patil Rachna Yogesh et al.], 2020)

2.6.3 Game-based Security Proofs:

An exact, mathematical method for proving the security of cryptographic systems are game-based security proofs. Under this approach, a protocol's security is seen as a "game" between an adversary (the attacker) and a challenger (symbolizing the protocol). The adversary wants to "win" the game by either forging a signature, deciphering a ciphertext, or posing as a user and so violating the protocol.

If the likelihood that any effective opponent may win this game is negligible—that is, so small that it can be disregarded for all practical purposes—the protocol is said to be secure. This method lets scientists measure security guarantees under well defined computational hardness assumptions, such as the hardness of factoring big numbers or solving discrete logarithm problems.

Usually involving a sequence of "games" or hybrid experiments each slightly dif-

ferent, game-based proofs show the adversary's success probability to remain negligible. These proofs clearly address adaptive attackers who might interact with the system in intricate ways, so offering a strong basis for cryptographic security.

Chapter 3

PROTOCOL DESIGN

This chapter presents the design of the proposed mutual authentication and key agreement protocol tailored for smart grid communication. It outlines the participating entities, assumptions, and step-by-step protocol phases ensuring secure and efficient data exchange.

3.1 Participating Entities

In the proposed mutual authentication and key agreement protocol, there are three main participants. Each plays an important role in ensuring secure communication in the smart grid environment:

3.1.1 Smart Meter (SM)

The Smart Meter is installed in the user's home or building. It regularly measures electricity usage and reports this data to the service provider. Since it operates in a remote and often unattended location, the Smart Meter needs strong security features to protect the data it sends and to verify who it is communicating with.

3.1.2 Service Provider (SP)

The Service Provider is the organization that supplies electricity and manages the billing system. It receives usage data from smart meters, processes it, and bills customers

accordingly. It also needs to authenticate the smart meter to ensure the data is coming from a trusted source and not from an attacker.

3.1.3 Trusted Authority (TA) or Control Center

The Trusted Authority is a secure and central entity responsible for initializing the system. It helps in generating and distributing cryptographic parameters, keys, and identities to both the Smart Meter and the Service Provider. It acts as the backbone of trust in the system. In some systems, this role may be played by a Control Center that oversees network-wide security and communication.

Together, these entities work in coordination to securely exchange messages, authenticate one another, and protect user data from potential threats in the smart grid environment.

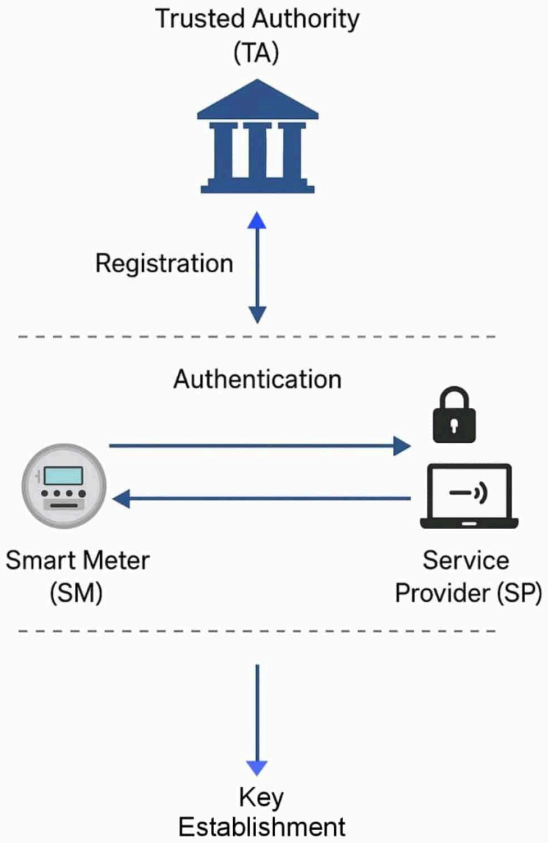

Figure 3.1: Overview of Registration, Authentication, and Key Establishment Between Smart Meter (SM), Service Provider (SP), and Trusted Authority (TA)

3.2 Assumptions

To design a secure and efficient mutual authentication and key agreement protocol, we make the following assumptions:

Cryptographic Security: The cryptographic algorithms used (e.g., hash functions, elliptic curve cryptography, AES) are assumed to be secure and resistant to known attacks.

Insecure Communication Channel: The communication between smart meters, service providers, and the trusted authority may pass through public or untrusted networks, and can be intercepted by attackers. Therefore, all messages must be protected through encryption and authentication.

Pre-shared or Registered Information: It is assumed that each entity (Smart Meter, Service Provider, and Trusted Authority) has some pre-shared or registered data such as identity, cryptographic keys, or public parameters, which are exchanged during a secure registration phase.

Resource Constraints on Smart Meters: Smart meters are considered to be resource-constrained devices with limited computational power, memory, and energy. Therefore, the proposed protocol is designed to be lightweight and efficient.

Trusted Authority (TA): The Trusted Authority is assumed to be a secure and trusted party responsible for user registration, key generation, and distribution. It is not compromised and always behaves correctly.

3.3 Design Goals

The main objective of the proposed protocol is to provide a secure and efficient communication framework for smart grid environments. The protocol is designed with the following goals:

Mutual Authentication: Ensure that the Smart Meter (SM) and the Service Provider (SP) can verify each other's identity, preventing unauthorized entities from participating in the communication.

Key Agreement: Establish a shared secret key between the SM and SP securely during the authentication phase, which will be used to encrypt future communication.

User Anonymity and Privacy: Protect the identity of users (e.g., smart meters) from being exposed to external observers or attackers, ensuring that user activities remain private.

Resistance to Common Attacks: The protocol is designed to resist various security threats such as replay attacks, man-in-the-middle (MITM) attacks, impersonation attacks, and denial-of-service (DoS) attacks.

Lightweight Computation: The protocol must be efficient and suitable for smart meters, which have limited processing power, memory, and energy. Lightweight cryptographic operations such as hashing and elliptic curve computations are preferred.

Forward Secrecy: Even if a long-term key is compromised, past session keys should remain secure.

3.4 Notations

Table 3.1: Description of Symbols Used in the Protocol

Symbol	Description
SM_i	i^{th} Smart Meter
SP_j	j^{th} Service Provider
ID_i	Identity of the smart meter i
r_1, r_2, r_3	Random numbers generated by the smart meter and service provider
Q_i	A value derived from the smart meter's identity and random value r_1 using HMAC
$salt_1$	A random value used in HMAC to ensure the freshness of the key
X_i	A value derived from r_2 and HMAC$_{salt_1}(ID_i \| r_1)$, used for encryption
$Enc(X_i)$	Encrypted value of X_i using a lightweight cipher or AES with a key derived from $H(ID_i \| r_1)$
SK	A session key shared between the smart meter and service provider, derived from the HMAC of r_2 and r_3

3.5 Proposed protocol

This protocol is designed to address the security and privacy concerns of smart meter data while ensuring efficient key establishment and mutual authentication between the smart meter (SM) and the service provider (SP). Our approach employs cryptographic techniques such as HMAC (Hash-based Message Authentication Code), lightweight encryption, and nonce-based mechanisms to safeguard against common security threats while minimizing computational and communication overhead. The protocol is divided into two main phases: Registration Phase and Authentication and Key Agreement Phase.

3.5.1 Registration Phase

In the registration phase, the smart meter (SM) generates a random value r_1 and a salt $salt_1$, which are used to create a secure identity binding. The smart meter computes a value Q_i using HMAC with the salt and the XOR of the smart meter's identity Id_i and r_1, as shown in Equation (1):

$$Q_i = \text{HMAC}_{salt_1}(ID_i \oplus r_1) \tag{3.1}$$

This ensures the confidentiality and integrity of the identity information while preventing potential replay attacks. The smart meter then sends the tuple $\{ID_i, Q_i, salt_1\}$ to the service provider SP_j over a secure channel. The service provider stores these values $\{ID_i, Q_i, r_1, salt_1\}$ in its database for future authentication interactions.

Table 3.2: Registration Phase

Smart Meter (SM_i)		**Service Provider** (SP_j)
Randomly generate r_1, $salt_1$		
Compute $Q_i = \text{HMAC}_{salt_1}(ID_i \oplus r_1)$		
	Sends $\{ID_i, Q_i, salt_1\}$ $- - - - - - \longrightarrow$	
		Store $\{ID_i, Q_i, r_1, salt_1\}$ in its database

3.5.2 Authentication and Key Agreement Phase

The authentication and key agreement phase starts when the smart meter SM_i wants to initiate communication with the service provider. The smart meter generates a random value r_2 and a nonce $nonce_1$ to ensure freshness and prevent replay attacks. It then

Table 3.3: Authentication and Key Agreement Phase

Smart Meter(SM_i)		**Service Provider**(SP_j)
Randomly generate r_2, $nonce_1$ Compute $X_i = r_2 \oplus \text{HMAC}_{salt_1}(ID_i \,\|\| r_1)$ $E_{nc}(X_i) = E_k(X_i)$ k = $H\,(ID_i \,\|\| r_1)$		
	Sends $\{ID_i, E_{nc}(X_i), nonce_1\}$ $-\!-\!-\!-\!-\!\longrightarrow$	
		Retrieve $\{Q_i,\ r_1,\ salt_1\}$ from database Decrypt $E_{nc}(X_i)$ using k $X_i = r_2 \oplus \text{HMAC}_{salt_1}(ID_i \,\|\| r_1)$ Validate $Q_i = \text{HMAC}_{salt_1}(ID_i \oplus r_1)$ Randomly generate r_3, $nonce_2$ Compute $SK = \text{HMAC}_{salt_1}(r_2' \,\|\| r_3)$ $Authh = \text{HMAC}_{SK}(ID_i \,\|\| r_2' \,\|\| r_3 \,\|\| nonce_1)$
	Sends $\{E_{nc}(r_3), Authh, nonce_2\}$ $\longleftarrow\!-\!-\!-\!-\!-\!-\!-$	
Decrypt $E_{nc}(r_3)$ using k Compute $SK = \text{HMAC}_{salt_1}(r_2 \,\|\| r_3)$ $Authh' = \text{HMAC}_{SK}(ID_i \,\|\| r_2 \,\|\| r_3 \,\|\| nonce_1)$ Validate $Authh = Authh'$, $nonce_2$ If valid, $SK = \text{HMAC}_{salt_1}(r_2 \,\|\| r_3)$ Session key established		

computes a new value X_i as:

$$X_i = r_2 \oplus \text{HMAC}_{salt_1}(ID_i \,\|\, r_1) \tag{3.2}$$

The value X_i is encrypted using a lightweight encryption algorithm, where the encryption key k is derived from $H\,(ID_i \,\|\, r_1)$, guaranteeing that only the intended service provider can decrypt the message.

The encrypted value $E_{\text{nc}}(X_i)$ along with ID_i and $nonce_1$ are sent to the service provider. The service provider SP_j retrieves the stored values Q_i, r_1, and $salt_1$ from its database. It then decrypts the received ciphertext $E_{\text{nc}}(X_i)$ using the same key k, derived from $H\,(ID_i \parallel r_1)$. Upon obtaining the value X_i, the service provider computes r_2' as:

$$r_2' = X_i \oplus \text{HMAC}_{\text{salt}_1}(ID_i \parallel r_1) \tag{3.3}$$

This calculation is used to check the authenticity of the message and whether the message is altered or not. The service provider also checks that Q_i matches the expected value derived from ID_i and r_1, ensuring that the smart meter's identity has not been tampered with.

To further secure the communication, the service provider generates a new random value r_3 and a second nonce $nonce_2$. It then computes a shared session key SK using:

$$SK = \text{HMAC}_{\text{salt}_1}(r_2' \parallel r_3) \tag{3.4}$$

This session key is used for future communication between the smart meter and the service provider. Additionally, the service provider generates an authentication code $Authh$ to guarantee that no message has been changed of the entire message. The service provider sends the encrypted value of r_3 , the authentication code $Authh$, and $nonce_2$ to the smart meter.

After getting the response from service provider, the smart meter decrypts r_3 using the same key k and computes the shared session key SK using the same method as the service provider. It then recalculates the authentication code $Authh$' and compares it to the received $Authh$. If both the authentication code and the nonces are valid, the session key is established, ensuring that both parties have authenticated each other and securely agreed on a shared key.

3.6 Protocol Correctness

The correctness of the protocol can be verified by checking whether the smart meters and service providers successfully authenticate each other, whether they derive the same session key, whether the nonces are fresh, and whether the cryptographic values are consistent and properly integrated.

3.6.1 Consistency in Decryption and Computation of r_2

Smart meter sends:

$$X_i = r_2 \oplus \text{HMAC}_{salt_1}(ID_i \,\|\, r_1)$$

$$E_{nc}(X_i) = E_k(X_i)$$

, where

$$k = H\left(ID_i \,\|\, r_1\right)$$

Service provider retrieves r_1, $salt_1$ and computes $k = H\left(ID_i \,\|\, r_1\right)$ and Decrypts

$$E_{nc}(X_i) \Rightarrow X_i$$

Recovers

$$r_2' = X_i \oplus \text{HMAC}_{\text{salt}_1}(ID_i \,\|\, r_1)$$

As both sides are using identical ID_i, r_1, $salt_1$ and HMAc function:

$$r_2' = r_2$$

So this is how Service provider correctly calculates the value of r_2 shared by smart meter.

3.6.2 Verification of Q_i

SP computes:

$$Q_i' = \text{HMAC}_{salt_1}(ID_i \oplus r_1)$$

As the Q_i here is calculated in the same way as it was calculated in registration phase, so this is how it verifies SM's identity.

3.6.3 Consistency in SK Derivation

Both entities compute:

$$SK = \text{HMAC}_{\text{salt}_1}(r_2' \parallel r_3)$$

SP sends Enc_{r_3} encrypted by $k = H\left(ID_i \parallel r_1\right)$ and calculates SK with known r_2, after decrypting it SM computes SK in same way. Hence both calculates same session key.

3.6.4 Validation of Authentication Token $Authh$

SP sends:

$$Authh = \text{HMAC}_{SK}(ID_i \parallel r_2' \parallel r_3 \parallel nonce_1)$$

SM verifies:

$$Authh' = \text{HMAC}_{SK}(ID_i \parallel r_2 \parallel r_3 \parallel nonce_1) \Rightarrow Authh' = Authh$$

As both side computes same SK and $Authh$ then authentication will also match.

Hence it proves the overall correctness of the protocol.

Chapter 4

IMPLEMENTATION

To verify the viability and efficacy of the suggested authentication and key agreement protocol in a real-world setting, it was implemented in Python. Python's cryptographic libraries are used for secure hash computation and lightweight encryption, and the implementation closely adheres to the protocol steps specified in the Registration and Authentication phases.

4.1 Tools and Libraries

- For the programming language, we chose Python 3.9+ because it has a rich set of libraries and is easy to read and use.
- Hash function and HMAC calculations were performed using the hmac and hashlib libraries.
- Sensitive data was encrypted and decrypted using a lightweight symmetric cypher (such as AES or a specially designed lightweight cypher).
- To ensure cryptographic strength, os.urandom() was used to implement random

number generation for nonces and random values (r_1, r_2, r_3).

4.2 Implementation Details

4.2.1 Registration Phase

- A random number, r_1, and a salt value, $salt_1$, are produced by the smart meter.
- Using $salt_1$ as the key over the XOR of the smart meter's ID and r_1, the value Q_i is calculated using HMAC.
- The service provider receives ID_i, Q_i, $salt_1$ securely from the smart meter.
- Together with r_1, the service provider saves these values for use in the authentication process later.

4.2.2 Authentication and Key Agreement Phase

- A new random number r_2 and nonce $nonce_1$ are produced by the smart meter.
- It calculates $X_i = r_2 \oplus \text{HMAC}_{salt_1}(ID_i \parallel r_1)$ and uses a lightweight cypher with a key derived from $H(ID_i||r_1)$ to encrypt X_i.
- The service provider receives ID_i, Enc(X_i), $nonce_1$ from the smart meter.
- The service provider decrypts Enc(X_i), recovers r_2, and retrieves stored parameters after receiving this message.
- By recalculating and validating Q_i and examining $nonce_1$ for replay attacks, the service provider confirms the authenticity.

- The shared session key $SK = HMAC_salt1(r_2||r_3)$ and the authentication code $Authh = HMAC_SK(ID_i||r_2||r_3||nonce_1)$ are calculated by the service provider after r_3 and $nonce_2$ are generated.

- Enc(r_3), Authh, $nonce_2$ are sent back to the smart meter.

- The smart meter calculates the shared key SK, decrypts Enc(r_3), and confirms $nonce_2$ and Authh.

- Both parties create a shared session key (SK) for secure communication after verification is successful.

4.3 Output and Validation

As seen in the Fig. 4.1, the program produces intermediate calculated values for every step.

[Registration Phase]
smart_meter_id (hex): 736d6172746d657465723031
r1: 81c462db283e74bc
salt: 6816e6ad70f08fbcf3d50230718057d5
Qi: d18296e471fc644569dcf343b15190916ed1bd06fa84593bce740b34b2a8a623
Registration Phase: IDi=736d6172746d657465723031, Qi=d18296e471fc644569dcf343b15190916ed1bd06fa84593bce740b34b2a8a623, r1=81c462db283e74bc, salt=6816e6ad70f08fbcf3d50230718057d5
[Authentication Phase]
[Sender: Smart Meter]
r2 (generated): 55d028ed8b6792a6
Salt: 6816e6ad70f08fbcf3d50230718057d5
smart_meter_id + r1: 736d6172746d657465723031 81c462db283e74bc
hmac(salt, smart_meter_id + r1): 35e67129ff395b5e460cba857ef59c49caa744ac754b2af693a2d4943e120c55
X_i (calculated): 060003060509 0c040704050e0c090f08
Encryption key (AES): b'5a8c2d29e9ef2b26'
Enc_Xi (encrypted): 3e12ddff7511570adadd88e5728bf5ac6452885def132b18046377cda1e64f250ec1f480ddd6775633d6d08cfc2b3e17
[Receiver: Service Provider]
Dec_Xi (decrypted): 060003060509 0c040704050e0c090f08
r2_prime (calculated): 03030e060702020f0f0a0300050705 0a040100080b0f080b07020f0c09030401
Qi (received): d18296e471fc644569dcf343b15190916ed1bd06fa84593bce740b34b2a8a623
Expected Qi (hmac(salt, xor(smart_meter_id, r1))): d18296e471fc644569dcf343b15190916ed1bd06fa84593bce740b34b2a8a623
[Debug: Service Provider] Inputs for SK (SP): r2_prime=03030e060702020f0f0a0300050705 0a040100080b0f080b07020f0c09030401, r3=a6e18e82964f88f8
[Debug: Service Provider] Session Key (SK): c55bf9ad5b21fea9c54b12cc3e9ca17e675f7a4c843833eb5fc7190fce6fb2e5
r3 (generated): a6e18e82964f88f8
SK (session key): c55bf9ad5b21fea9c54b12cc3e9ca17e675f7a4c843833eb5fc7190fce6fb2e5
Authh (calculated): 0cee3ef6bd944663ae25ff2e105789a30d18183454679beaa0afa79f9b03213c
Enc_r3 (encrypted): 37a279722522d68b1b15bf51517222ae0ec1f480ddd6775633d6d08cfc2b3e17
[Debug: Smart Meter] Inputs for SK_SM (Smart Meter): r2=55d028ed8b6792a6, Dec_r3=a6e18e82964f88f8
[Debug: Smart Meter] Session Key (SK_SM): b19a73acceb80e12a32e3064e6447e806c4c89f179aa383d9114d82aae7ea878
[Debug: Smart Meter] Inputs for Authh_prime: IDi=736d6172746d657465723031, r2=55d028ed8b6792a6, Dec_r3=a6e18e82964f88f8, nonce1=4ff59523
[Debug: Smart Meter] Authh_prime: 0cee3ef6bd944663ae25ff2e105789a30d18183454679beaa0afa79f9b03213c
[Debug: Service Provider] Inputs for Authh: IDi=736d6172746d657465723031, r2_prime=03030e060702020f0f0a0300050705 0a040100080b0f080b07020f0c09030401, r3=a6e18e82964f88f8, nonce1=4ff59523
[Debug: Service Provider] Authh: 0cee3ef6bd944663ae25ff2e105789a30d18183454679beaa0afa79f9b03213c
[Sender: Smart Meter Verification]
Dec_r3 (decrypted): a6e18e82964f88f8
Session Key (SK_SM): b19a73acceb80e12a32e3064e6447e806c4c89f179aa383d9114d82aae7ea878
Authh_prime (calculated): 0cee3ef6bd944663ae25ff2e105789a30d18183454679beaa0afa79f9b03213c
Authh (from SP): 0cee3ef6bd944663ae25ff2e105789a30d18183454679beaa0afa79f9b03213c
Authentication Successful! Session Key: b19a73acceb80e12a32e3064e6447e806c4c89f179aa383d9114d82aae7ea878

Figure 4.1: Snapshot of protocol execution with computed values and session key

Chapter 5

EXPERIMENTS AND RESULTS

5.1 Security Analysis

This section offers a thorough examination of the suggested protocol's resilience to different possible security risks. The objective is to show that the protocol not only accomplishes secure key agreement and mutual authentication, but also fends off known attacks. The protocol's resilience against insider threats, forward secrecy, and user privacy are also investigated.

5.1.1 Informal Security Analysis

Mutual Authentication: The goal is to make sure that the service provider (SP_j) and smart meter (SM_i) authenticate one another. According to the analysis:

1. During the registration phase, the authentication credential is $Q_i = \mathrm{HMAC}_{salt_1}(ID_i \oplus r_1)$.

2. In the authentication stage, SP_j uses ID_i, r_1, and $salt_1$ that it has retrieved from its database to validate Qi.

    - $Authh = \text{HMAC}_{SK}(ID_i \parallel r_2 \parallel r_3 \parallel nonce_1)$ confirms SP_j's authenticity to SM_i, and $Authh'$ is used to confirm SM_i.

3. Without knowledge of r_1, r_2, or SK, an adversary cannot forge Qi or Authh. By guaranteeing that only authorised entities are able to calculate and validate Qi and Authh, the protocol accomplishes mutual authentication.

By guaranteeing that only authorised entities are able to calculate and validate Qi and Authh, the protocol accomplishes mutual authentication.

Session Key Security: Making sure the session key ($SK = \text{HMAC}_{salt_1}(r_2 \parallel r_3)$) is identical to a random value and shared only between SM_i and SP_j is the goal. According to the analysis:

1. The unpredictability of SK is guaranteed by the randomness of r_2 and r_3.
2. An adversary cannot determine r_2 or r_3 without cracking the encryption or HMAC, even if they manage to intercept Enc(X_i) or Enc(r_3).
3. The ROM's hash function (HMAC) makes sure that SK can't be computationally distinguished from a random value.

Under the suggested protocol, the session key is safely created and kept private.

Resistance to Replay Attacks: The objective is to prevent adversaries from reusing previous messages to pose as trustworthy organisations. According to the analysis:

1. Each session is guaranteed to be fresh through the use of nonces ($noce_1$ and $nonce_2$).

2. Enc(X_i) or Enc(r_3) cannot be replayed since:
    - $nonce_1$ is checked against a replay attack list.
    - New random values (r_2, r_3, $nonce_1$) are necessary for SK and Authh.
3. The session will end if the calculated $Authh$ or $Authh$' differs.

By using new random values and nonces, the protocol successfully thwarts replay attacks.

Resistance to Man-in-the-Middle (MITM) Attacks: Making sure the adversary cannot intercept or change messages covertly is the goal.

1. Enc(X_i) and Enc(r_3) safeguard the confidentiality of transmitted data, according to the analysis.
2. Message integrity is guaranteed by HMACs (such as $Authh$ and $Authh'$).
3. Without the necessary cryptographic secrets (r_1, r_2, r_3, $salt_1$, or SK), the adversary is unable to produce legitimate Qi, $Authh$, or $Authh'$.

By ensuring message confidentiality, integrity, and authenticity, the protocol fends off MITM attacks.

Resistance to Key Compromise Impersonation (KCI): The goal is to stop an adversary from pretending to be one party in the event that the session key is stolen. The analysis demonstrates that:

1. The adversary cannot pretend to be SM_i or SP_j even if SK is compromised because:
    - r_1, r_2, and r_3 are independently random for every session and are necessary for the generation of Q_i and $Authh$.

- $salt_1$ and ID_i are kept secret.

The protocol uses independent random values for every session to provide KCI resistance.

Brute Force and Dictionary Attacks:

1. Brute force attacks are prevented by using HMAC and encryption with a long enough key.
2. Precomputed dictionary attacks on Qi are prevented by salt values ($salt_1$).

Denial-of-Service (DoS) Attacks:

1. Replayed or invalid requests are not processed excessively thanks to nonce validation.
2. To lessen the effect of DoS attacks, the protocol ends instantly if any validation step is unsuccessful.

5.1.2 Formal Security Analysis

We use BAN logic, a formal technique for analysing authentication protocols, to demonstrate the security of our suggested protocol. The objective is to confirm that the service provider (SP_j) and the smart meter (SM_i) securely agree on a session key and mutually authenticate one another. **Step 1: Extract Protocol Messages:**

- **Message 1:** $SM_i \rightarrow SP_j : ID_i$, Enc($X_i$), $nonce_1$
- **Message 2:** $SP_j \rightarrow SM_i$: Enc(r_3), $Authh$, $nonce_2$

After this exchange:

- SM_i decrypts Enc(r_3) and computes the session key SK.
- Both parties validate Authh.
- If all checks pass, the session key SK is securely established.

Step 2: BAN Logic Notations:

Notation	Meaning
$P \mid\equiv X$	P believes X
$P \mid\sim X$	P sees message X
$P \Rightarrow X$	P has jurisdiction over X
$\#(X)$	X is fresh (not replayed)
$P \overset{K}{\leftrightarrow} Q$	P and Q share secret key K
$\langle X \rangle_K$	Message X is encrypted using key K
$P \triangleleft X$	P once said X

Initial Assumptions

- Nonce Freshness:

$$SM_i \mid\equiv \#(\text{nonce}_1), \quad SP_j \mid\equiv \#(\text{nonce}_2)$$

(Smart meter and Service provider believes that $nonce_1$ and $nonce_2$ are fresh respectively.)

- Key Agreement:

$$SM_i \mid\equiv SP_j \Rightarrow SK, \quad SP_j \mid\equiv SM_i \Rightarrow SK$$

(Smart meter believes Service provider has the right to establish SK, similarly Service provider believes Smart meter has the right to establish SK.)

- Message Meaning:

$$SP_j \mid\sim \langle X_i \rangle_K, \quad SM_i \mid\sim \langle r_3 \rangle_K$$

(Smart meter and Service provider sees an encrypted message.)

Step 3: Applying BAN Logic Rules

Message 1: $SM_i \rightarrow SP_j$

- SP_j sees $\langle X_i \rangle_K$:

$$SP_j \mid\sim \langle X_i \rangle_K$$

- By Message Meaning Rule:

$$SP_j \mid\equiv SM_i \mid\sim X_i$$

- Since SP_j believes $\#(\text{nonce}_1)$:

$$SP_j \mid\equiv \#(X_i)$$

Message 2: $SP_j \rightarrow SM_i$

- SM_i sees $\langle r_3 \rangle_K$:

$$SM_i \mid\sim \langle r_3 \rangle_K$$

- By Message Meaning Rule:

$$SM_i \mid\equiv SP_j \mid\sim r_3$$

- Since SM_i believes $\#(\text{nonce}_2)$:

$$SM_i \mid\equiv \#(r_3)$$

Message 3: Session Key Agreement

Both parties compute:

$$SK = \text{HMAC}_{salt1}(r_2 \parallel r_3)$$

Since r_2 and r_3 are fresh:

$$SM_i \mid\equiv SP_j \mid\equiv SK, \quad SP_j \mid\equiv SM_i \mid\equiv SK$$

Step 4: Final Security Proof

- **Goal 1: Mutual Authentication**

$$SM_i \mid\equiv SP_j \mid\equiv SK, \quad SP_j \mid\equiv SM_i \mid\equiv SK$$

- **Goal 2: Key Secrecy**

  SK is only known to SM_i and SP_j; cannot be derived by an adversary.

- **Goal 3: Replay Attack Resistance**

  Fresh nonces nonce_1 and nonce_2 prevent reuse of old messages.

- **Goal 4: Man-in-the-Middle Resistance**

  All messages are either encrypted or authenticated using HMAC, ensuring confidentiality and integrity.

5.2 Performance Analysis

We examine the performance of our suggested mutual authentication and key agreement protocol using computation and communication costs in order to assess its usefulness and efficiency.

Table 5.1: Cryptographic Operations

Notations	**Explanation**	**Value in ms**
T_{HO}	Hash Operation	0.005ms
T_{PM}	Point Multiplication	2.226ms
T_{par}	Asymmetric Bilinear Pairing Operation	5.8110 ms
T_{EPA}	ECC Point Addition	0.0288 ms
T_{AE}	Asymmetric Encryption\Decryption	1ms
T_{SE}	Symmetric Encryption\Decryption	0.0046ms
T_{HMAC}	HMAC Operation	0.1ms

5.2.1 Computation Cost

In our scheme, the service provider (SP_j) and the smart meter (SM_i) split up the work. Four HMAC calculations (T$_H MAC$), one hashing operation (T_H), one decryption (T_d), and one encryption (T_e) are all carried out by the smart meter. A comparable set of tasks is carried out by the service provider. The protocol's overall computational cost, when added together, is

$$8T_{HMAC} + 2T_H + 2T_d + 2T_e$$

According to Table 5.2 and Fig. 5.1 of the computational cost comparison, the protocol's total computational time is 1.060 ms.

Table 5.2: Computation Cost

Schemes	**Computation Cost**	**Accumulated Computation Cost**
[2]	$T_{PM} + 7T_{HO}$	2.2536 ms
[8]	$4T_{PM} + 7T_{HO}$	8.9201 ms
[10]	$8T_{HO} + 3T_{PM}$	4.076 ms
[11]	$9T_{PM} + 2T_{par}$	61.80 ms
[12]	$27T_{PM}$	45.159 ms
[13]	$10T_{AE} + 12T_{HO} + 10T_{EPA}$	11.2878 ms
[1]	$17T_{HO} + 2T_{AE} + 2T_{SE} + 5T_{PM}$	54.3118 ms
Proposed	$8T_{HMAC} + 2T_{HO} + 4T_{SE}$	1.060 ms

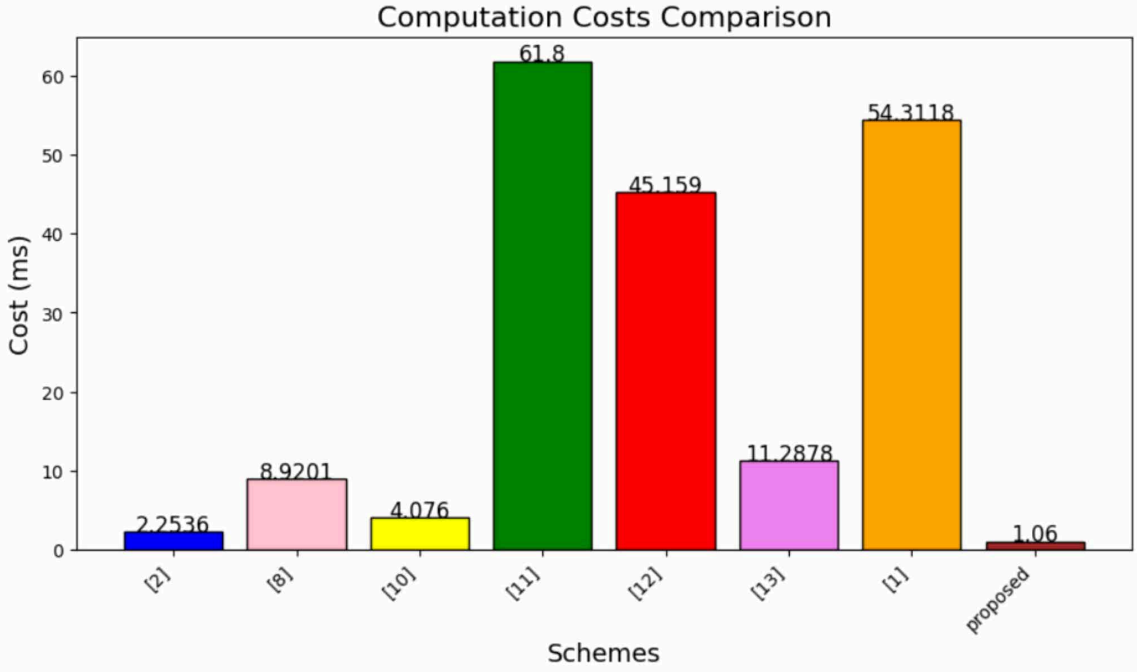

Figure 5.1: Computation Cost

5.2.2 Communication Cost

By employing effective cryptographic techniques and carefully planning the message exchanges, our suggested scheme reduces communication overhead. Identifiers, encrypted keys, and authentication codes are all part of the communication between the smart meter (SM_i) and the service provider (SP_j); these are all designed to reduce bit-length while preserving functionality and security. Table 5.3 and Fig. 5.2 compare the communication

costs of various schemes. The protocol's total communication cost is 96 bytes, or 768 bits.

Table 5.3: Communication Cost

Schemes	**Communication Cost**
[2]	1152 bits
[8]	1152 bits
[10]	992 bits
[11]	1504 bits
[12]	2816 bits
[13]	2080 bits
[1]	576 bits
Proposed	768 bits

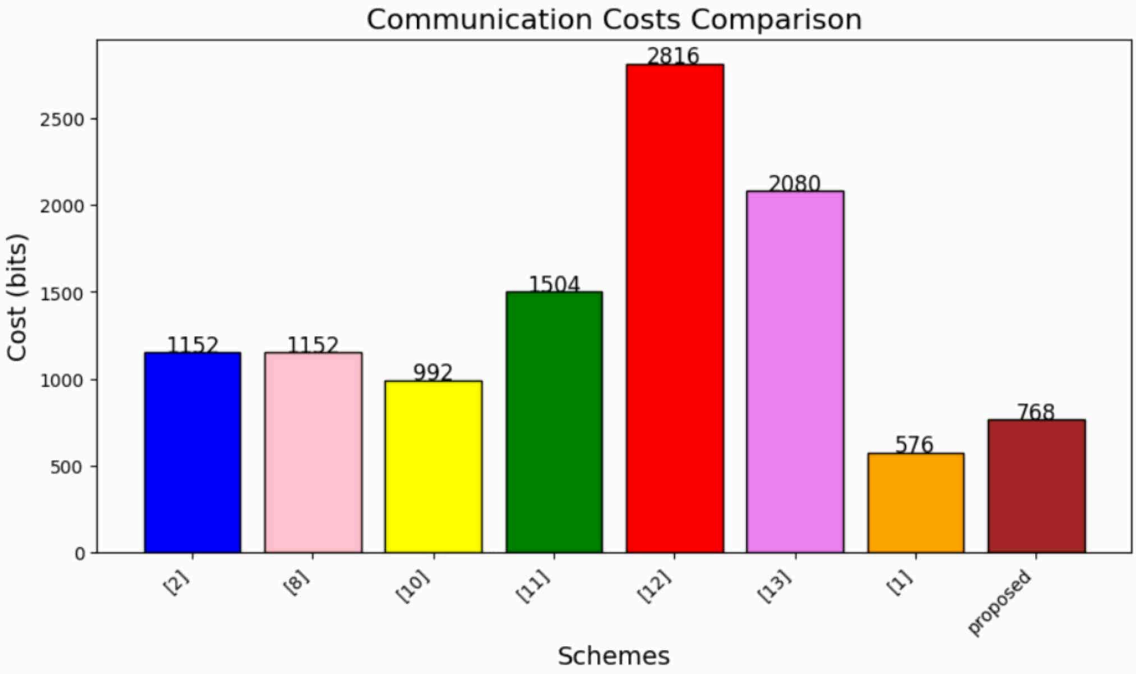

Figure 5.2: Communication Cost

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this work, we proposed a robust mutual authentication and key agreement protocol for secure communication between smart meters and service providers. The proposed protocol achieves low computational cost and is energy-efficient, with a computation time of only 1.060 ms — significantly lower compared to existing schemes. We also calculated the communication cost, which is 768 bits, making the protocol lightweight and suitable for resource-constrained devices such as smart meters. In addition, the protocol provides various security features, including resistance to several attacks such as replay and man-in-the-middle (MITM) attacks.

6.2 Future Work

- **Testing and Deployment in the Real World** To assess the suggested protocol's real performance, security, and adaptability with operational devices and network

conditions, it can be tested in a real smart grid environment.

- **Adding IoT Devices** This protocol can be expanded in the future to accommodate additional IoT elements in the smart grid, like home sensors and smart appliances, which might have different resource and communication constraints.
- **Cryptography Resistant to Quantum** In order to guarantee the long-term security of key exchange and authentication, post-quantum cryptography techniques must be investigated as quantum computing advances.
- **Optimisation of Energy and Resources** For devices with very limited resources, the protocol can be further optimised to use less memory and less energy.
- **Integration of Multi-factor Authentication** The level of security in smart grid authentication can be raised by investigating additional layers like biometric verification or token-based techniques.
- **Assistance with Mobility and Roaming** improving the protocol to allow devices to switch between service providers or network segments without violating security assurances.

Bibliography

[1] Disuja Kumari, Karan Singh, Lightweight secure authentication and key agreement technique for smart grid, Peer-to-Peer Networking and Applications, vol. 17, pp. 451–478, 2024.

[2] Ram Baksh, Samiulla Itoo, Musheer Ahmad, A comprehensive and secure scheme for privacy-preserving smartm eter data aggregation in the smart grid, Sustainable Energy,Grids and Networks, vol. 39, pp. 101461, 2024.

[3] Shehzad Ashraf Chaudhry, Hosam Alhakami, Abdullah Baz, Fadi Al-Turjman, Securing Demand Response Management: A Certificate-Based Access Control in Smart Grid Edge Computing Infrastructure, IEEE, 2020.

[4] Hafiz Muhammad Sanaullah Badar, Salman Qadri, Salman Shamshad, Muhammad Faizan Ayub, Khalid Mahmood, Neeraj Kumar, An Identity Based Authentication Protocol for Smart Grid Environment Using Physical Uncloneable Function, IEEE, vol. 12, no. 5, 2021.

[5] Zhang, Y., Deng, R. H., Liu, X., Towards attributes-based authentication for mobile devices. IEEE Transactions on Information Forensics and Security, vol. 10(9), pp. 1847–1861, 2015.

[6] Mohammad Kamrul Hasan, Md Mehedi Hasan, Anil Kumar Budati, Shayla Islam, Nurhizam Safie, Fatima Rayan Awad Ahmed, Khairul Azmi Abu Bakar, Nissrein Babiker Mohammed Babiker, Taher M. Ghaza, A hybrid key agreement scheme uti-

lized elliptic curve Diffie-Hellman for IoT based advanced metering environment, Earth Science Informatics, vol. 17, pp. 2447–2460, 2024.

[7] Tanksale, V., Efficient Elliptic Curve Diffie–Hellman Key Exchange for Resource-Constrained IoT Devices. Electronics, vol. 13(18), pp. 3631, 2024.

[8] Akber Ali Khan, Vinod Kumar, Musheer Ahmad, An elliptic curve cryptography based mutual authentication scheme for smart grid communications using biometric approach, Journal of King Saud University - Computer and Information Sciences, vol. 34, pp. 698-705, 2022.

[9] Waseem Akram, Khalid Mahmood, Xiong Li, Mazhar Sadiq, Zhihan, Shehzad Ashraf Chaudhry, An energy-efficient and secure identity based RFID authentication scheme for vehicular cloud computing, Computer Networks Vol. 217, pp. 109335, 2022.

[10] S. M. S. Badar, K. Mahmood, W. Akram, Z. Ghaffar, M. Umar, A.K. Das, Secure authentication protocol for home area network in smart grid-based smart cities, Computers and electrical engineering, vol. 108, pp. 108721, 2023.

[11] V. Odelu, A. K. Das, S. Kumari, X. Huang, M. Wazid, Provably secure authenticated key agreement scheme for distributed mobile cloud computing services, Future generation computer systems, vol. 68, pp. 74-88, 2017.

[12] Zhe Xia, Tao Liu, Jingjing Wang, Shi chen, A secure and efficient authenticated key exchange scheme for smart grid, Heliyon, vol. 9, pp. 17240, 2023.

[13] Prarthana J. Mehta, balu L. Parne, sankita J. Patel, SE-LAKAF: security enhanced lightweight authentication and key agreement framework for smart grid network, Peer-to-peer networking and applications, vol. 16, pp. 1513–1535, 2023.

[14] A.K. Das, B. Bera, S. Saha, A.V. Vasilakos.: blockchain- based access control protocol in IoT-enabled smart- grid system. IEEE Internet of Things, vol. 8(7), pp. 5744-5761, 2020.

[15] S.A. Chaudhry, H. Alhakami, A. Baz, F. Al-Turjman, "Securing demand response management: A certificate-based access control in smart grid edge computing infrastructure," IEEE Access, vol. 8, pp. 101235–101243, 2020.

[16] D. Abbasinezhad-Mood, A. Ostad-Sharif, M. Nikooghadam, S.M. Mazi nani, A secure and efficient key establishment scheme for communica tions of smart meters and service providers in smart grid, IEEE Trans. Ind. Inform., vol. 16, (3), pp. 1495–1502, 2019.

[17] J. Xia, Y. Wang, Secure key distribution for smart grid, IEEE Trans. Smart Grid, vol. 3, (3), pp. 1437–1443, 2012.

[18] J.-L. Tsai, N.-W. Lo, Secure anonymous key distribution scheme for smart grid, IEEE Trans. Smart Grid, vol. 7, (2), pp. 906–914, 2015.

[19] K. Mahmood, W. Akram, A. Shafiq, I. Altaf, M.A. Lodhi, S.H. Islam, An enhanced and provably secure multi-factor authentication scheme for Internet-of-Multimedia-Things environments, Comput. Electr. Eng., vol. 88, pp. 106888, 2020.

[20] H. Hammami, S.B. Yahia, M.S. Obaidat, A novel efficient and lightweight authentication scheme for secure smart grid communication systems, J. Supercomput.,pp. 1–17, 2022.

[21] J. Srinivas, A.K. Das, X. Li, M.K. Khan, M. Jo, Designing anonymous signature-based authenticated key exchange scheme for Internet of Things-enabled smart grid systems, IEEE Trans. Ind. Inform., vol. 17, (7), pp. 4425–4436, 2020.